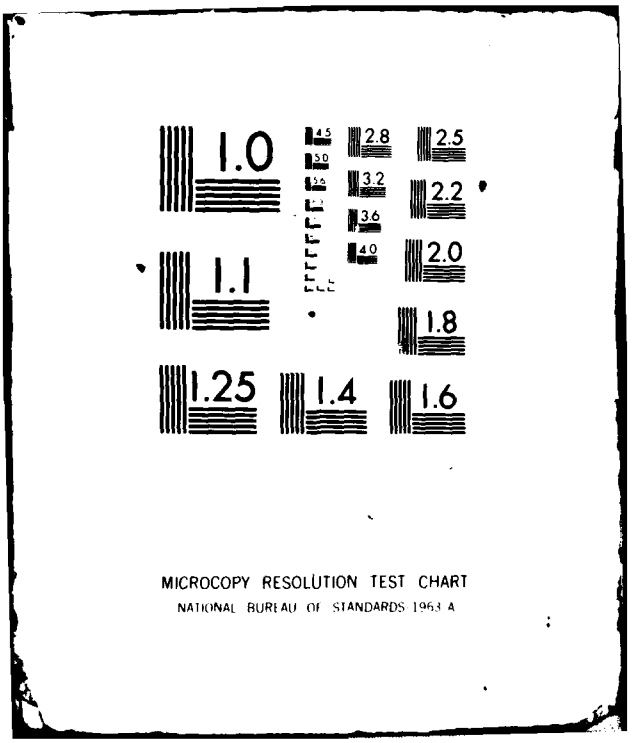
AD-A098 418 MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTE--ETC F/G 12/1
ON THE REPRESENTATION OF ANGULAR VELOCITY AND ITS EFFECT ON THE--ETC(U)
MAR 81 W M SILVER N00014-77-C-0389
UNCLASSIFIED AI-M-622 NL

END
DATE FILMED
4 81
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

LEVEL II

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AIM 622
2. GOVT ACCESSION NO.: AD-A098418
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): On the Representation of Angular Velocity and its Effect on the Efficiency of Manipulator Dynamics Computation.
5. TYPE OF REPORT & PERIOD COVERED: Memorandum rept.
6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): William M. Silver
8. CONTRACT OR GRANT NUMBER(s): N00014-77-C-0389
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Artificial Intelligence Laboratory, 545 Technology Square, Cambridge, Massachusetts 02139
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
11. CONTROLLING OFFICE NAME AND ADDRESS: Advanced Research Projects Agency, 1400 Wilson Blvd, Arlington, Virginia 22209
12. REPORT DATE: March 1981
13. NUMBER OF PAGES: 28
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Office of Naval Research, Information Systems, Arlington, Virginia 22217
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:
16. DISTRIBUTION STATEMENT (of this Report): Distribution of this document is unlimited.

AI-M-622

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
18. SUPPLEMENTARY NOTES: None
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): Robotics, Lagrangian dynamics, Manipulators, Newton-Euler dynamics
20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

Recently there has been considerable interest in efficient formulation of manipulator dynamics, mostly due to the desirability of real-time control or analysis of physical devices using modest computers. The inefficiency of the classical Lagrangian formulations is well known, and this has led researchers to seek alternative methods. Several authors have developed a highly efficient formulation of manipulator dynamics based on the Newton-Euler equations, and there may be some confusion as to the source of this efficiency.

DTIC ELECTE MAY 4 1981 S D

407483

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

81 4 30 226

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AD A098418

DTIC FILE COPY

20. This paper shows that there is in fact no fundamental difference in computational efficiency between Lagrangian and Newton-Euler formulations. The efficiency of the above-mentioned Newton-Euler formulation is due to two factors: the recursive structure of the computation and the representation chosen for the rotational dynamics. Both of these factors can be achieved in the Lagrangian formulation, resulting in an algorithm identical to the Newton-Euler formulation. Recursive Lagrangian formulation, resulting in an algorithm identical to the Newton-Euler formulation. Recursive Lagrangian dynamics has been discussed previously by Hollerbach. This paper takes the final step by comparing in detail the representations that have been used for rotational dynamics and showing that with a proper choice of representation the Lagrangian formulation is indeed equivalent to the Newton-Euler formulation.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

A.I. Memo No. 622 March, 1981

On the Representation of Angular Velocity and its Effect on the Efficiency of Manipulator Dynamics Computation

William M. Silver

Abstract

Recently there has been considerable interest in efficient formulations of manipulator dynamics, mostly due to the desirability of real-time control or analysis of physical devices using modest computers. The inefficiency of the classical Lagrangian formulation is well known, and this has led researchers to seek alternative methods. Several authors have developed a highly efficient formulation of manipulator dynamics based on the Newton-Euler equations, and there may be some confusion as to the source of this efficiency. This paper shows that there is in fact no fundamental difference in computational efficiency between Lagrangian and Newton-Euler formulations. The efficiency of the above-mentioned Newton-Euler formulation is due to two factors: the recursive structure of the computation and the representation chosen for the rotational dynamics. Both of these factors can be achieved in the Lagrangian formulation, resulting in an algorithm identical to the Newton-Euler formulation. Recursive Lagrangian dynamics has been discussed previously by Hollerbach. This paper takes the final step by comparing in detail the representations that have been used for rotational dynamics and showing that with a proper choice of representation the Lagrangian formulation is indeed equivalent to the Newton-Euler formulation.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's artificial intelligence research is provided in part by the Office of Naval Research under Office of Naval Research contract N00014-77C-0389.

Table of Contents

1. Background

Manipulator dynamics concerns the relationship between the motion of a mechanical kinematic chain of linkages and the forces applied by its actuators. For some problems, such as simulation, the forces are known and it is desired to compute the resulting motion. In other cases, such as the important area of real-time control, the desired motion is known and the forces necessary to achieve that motion must be computed. In either case, for a given model of a kinematic chain an exact solution can be found, within the framework of Newtonian mechanics. The former case reduces to a system of non-linear second order differential equations, which can be solved numerically. The latter case is easier-- the required forces can be expressed directly in terms of the known position, velocity and acceleration of the chain.

In this paper we restrict the discussion to open-loop kinematic chains, composed of rigid links connected by joints that allow relative motion of the links. We assume that each joint has only one degree of freedom, either rotational or translational. Multiple rotational degrees of freedom can be modelled by links of zero mass and length.

1.1 Lagrangian Generalized Coordinates

In Newton's original formulation of mechanics, the relationship between forces acting on bodies and the resulting accelerations is described using cartesian coordinate systems. There are other, equivalent ways to describe the dynamics of a system of bodies. One such method was invented by Lagrange, using what are known as generalized coordinates. Generalized coordinates are any convenient set of variables that completely define the position of a system of bodies. The Lagrange equation

$$\tau_i = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} \tag{1.1}$$

describes the relationship between the corresponding generalized forces acting on the bodies and the kinetic and potential energy of the system. Here q_i is a generalized coordinate, τ_i is the corresponding generalized force, and the dot indicates differentiation with respect to time. L is the Lagrangian—the difference between the total kinetic energy and the total potential energy of the system: $L = K - P$.

While the Lagrange equation must yield the same numerical results as direct application of Newton's laws, either approach may be more convenient than the other in a given situation, or may provide greater insight into the physics of the problem.

1.2 The Uicker/Kahn Formulation

A *kinematic* chain has a natural set of coordinates that completely specify its position—the joint variables q_i (angles for rotational joints and distances for sliding joints). The q_i satisfy the requirements for generalized coordinates. Furthermore, they can be measured directly by the manipulator and the corresponding generalized forces (torques for rotational joints and ordinary force for sliding joints) are just what can be controlled. It is not surprising, then, that the pioneering work of Uicker [1] and Kahn [2] on the dynamics of mechanical linkages made use of the Lagrangian method.

From the standpoint of the present discussion, the important feature of the work of Uicker and Kahn is their use of 4×4 rotation/translation matricies W_i to represent the position and motion of the kinematic chain. A coordinate system is attached to and moves with each link. The matrix W_i transforms the components of a vector with respect to link i coordinates to its components with respect to a fixed (inertial) coordinate system. The position and motion of the chain is described by the W_i's and their time derivatives, which are in turn functions of the q_i's and their time derivatives.

Once the kinetic and potential energy of the chain is expressed in terms of the W_i's and their derivatives, it is a straightforward matter to apply the Lagrange equation and find the generalized forces. The

result looks like

$$\tau_i = \sum_{j=i}^{n} tr\left(\frac{\partial W_j}{\partial q_i} J_j \ddot{W}_j^T\right), \tag{1.2}$$

where τ_i is the generalized force applied to the ith joint, J_j is the 4×4 inertia matrix of the jth link in that link's coordinates, and tr is the trace operation. The gravity term is omitted here because it is not important for this paper, although in general it must of course be included.

It has been observed by many authors that evaluating (1.2) directly as written requires time proportional to the fourth power of the number of links. Hollerbach [3] has determined that for 6 links well over 100,000 adds and multiplies would be needed to compute all of the τ_i's, and Luh et al. [4] report that a Fortran program running on a PDP-11/45 took nearly 8 seconds to compute them. Since a real-time control system would have to repeat this calculation at a rate on the order of 60 Hz, until recently it had been believed that a manipulator could not be controlled by direct real-time calculation of the actuator forces, without introducing approximations or using lookup tables.

The inefficiency of the original Uicker/Kahn formulation, as well as other reasons, have led researchers to look for alternative formulations of manipulator dynamics. The most successful of these has been the Newton-Euler approach.

1.3 The Newton-Euler Approach

In order to apply Newton's laws to objects which are not point masses, we consider such objects to be composed of a large number of point masses bound together by effectively infinite internal forces. The laws governing these so-called "rigid bodies" may be derived from Newtonian mechanics [5]. The key feature is that the description of motion is broken up into two independent components—linear motion of the center of mass (or other suitable point) and rotation of the body about that point. The total vector force acting on the body is related to the acceleration of the center of mass by Newton's second law: $F = m\dot{v}$. The total vector moment (torque) about the center of mass is related to the angular velocity and

angular acceleration of the body by Euler's equation

$$\underset{\sim}{N} = \underset{\approx}{I} \cdot \dot{\underset{\sim}{\omega}} + \underset{\sim}{\omega} \times (\underset{\approx}{I} \cdot \underset{\sim}{\omega}). \tag{1.3}$$

The Euler equation follows directly from the laws of rotating reference frames and the following definitions:

ω	angular velocity
$\underset{\approx}{I}$	inertia tensor
$\underset{\sim}{L} = \underset{\approx}{I} \cdot \underset{\sim}{\omega}$	angular momentum
$\underset{\sim}{N} = \dfrac{d\underset{\sim}{L}}{dt}$	moment (torque).

These definitions are analogous to those found in the case of linear motion, except that inertia is a second rank tensor instead of a scalar, since angular momentum is not in general parallel to angular velocity. If we use d^*/dt to indicate differentiation with respect to the rotating reference frame, we then have:

$$\begin{aligned} \underset{\sim}{N} = \frac{d\underset{\sim}{L}}{dt} &= \frac{d^*\underset{\sim}{L}}{dt} + \underset{\sim}{\omega} \times \underset{\sim}{L} \\ &= \frac{d^*}{dt}(\underset{\approx}{I} \cdot \underset{\sim}{\omega}) + \underset{\sim}{\omega} \times (\underset{\approx}{I} \cdot \underset{\sim}{\omega}) \\ &= \underset{\approx}{I} \cdot \dot{\underset{\sim}{\omega}} + \underset{\sim}{\omega} \times (\underset{\approx}{I} \cdot \underset{\sim}{\omega}). \end{aligned}$$

To apply the Newton-Euler equations to a kinematic chain, the following procedure may be used.

1. The base of the chain is either fixed or its motion is known. Starting from the base and working outwards, and using the known geometry of the chain, $\dot{\underset{\sim}{v}}_i$, $\underset{\sim}{\omega}_i$, and $\dot{\underset{\sim}{\omega}}_i$ of link i may be found in terms of the q_j, $\dot{q}_j$, and $\ddot{q}_j$ of the preceeding joints.
2. The total vector force $\underset{\sim}{F}_i$ and the total vector moment $\underset{\sim}{N}_i$ acting on each link may now be determined using the Newton/Euler equations.
3. The total force computed in step 2 is the vector sum of the forces exerted on the link by its neighbors at the joints, and the force of gravity. The total moment is the vector sum of the pure moments exerted on the link by its neighbors, and the moments generated by the forces exerted

by its neighbors. Thus if the force and moment acting at one end of a link are known, we can use the totals from step 2 and the known force of gravity to solve for the force and moment acting at the other end. These are equal and opposite to the force and moment exerted by the given link on its neighbor, by Newton's third law. Thus, if the force and moment exerted by the environment on the terminal link (e.g. the hand) are known, we can proceed down the chain to the base and determine the force and moment acting at each joint.

4. For rotational joints, the vector moment determined in step 3 is projected along the axis of rotation to yield the joint torque. For sliding joints, the vector force from step 3 is projected along the sliding axis to yield the joint force. The other components of the force and moment are generated by the structure and bearings of the device.

It is clear that many details must be filled in before the above procedure can actually be applied to a kinematic chain. We must have conventions for defining the geometry of the chain and specifying how the joint variables are to be measured, and coordinate systems that allow the vector and tensor quantities to be specified. The transformation required by step 1 must be worked out, and the operations specified by the other steps must be written down in detail. The efficiency of the resulting computation will depend on how these issues are resolved.

Recently a number of authors have been interested in the Newton-Euler approach, partly due to perceived problems with the Lagrangian formulation, as mentioned above. Stepanenko and Vukobratovic [6] worked out the details in connection with work on understanding the dynamics of human limbs. One of their main goals was to develop a computer program that could perform the tedious mathematical manipulations necessary to set up the equations of motion from a description of the kinematic chain. They rejected the Lagrange equation because of the differentiations it requires—there are severe problems associated with numerical differentiation, as they pointed out. These problems can be avoided, however, by deriving the differential equation of motion for an arbitrary kinematic chain, as is done in section 3 for the open-loop class. Numerical values for a specific device are then substituted, but at this point all of

the necessary differentiations have been done symbolically. The general solution can readily be found by hand, and only needs to be done once.

Stepanenko and Vukobratovic were not concerned with the efficiency of the computation. Their formulation was revised by Orin *et al.* [7] in connection with the control of robot legs. They improved upon the efficiency of Stepanenko and Vukobratovic by referring the forces and moments to coordinate systems attached to the links instead of fixed coordinates. They also noticed that the sequential nature of the computation (iterating from the base to the tip to determine the motion of the chain, and then from the tip to the base to determine the forces, as described above) seemed to reduce the computation time and storage requirements. They speculated that such a recursive procedure might be more efficient in general, but did not draw any conclusions. Armstrong [8] and Luh et al. [4] paid close attention to computational efficiency and confirmed these suspicions. They pointed out that the Newton-Euler formulation leads to an algorithm where the computation time grows linearly with the number of links, as opposed to the quartic behaviour of the original Lagrangian formulation. They further improved the efficiency by referring the linear and angular velocities and accelerations, as well as the forces and moments, to link coordinates. In addition, the need for efficiency produced a formulation which is simpler in many ways. For example, the three coordinate systems attached to each link by Stepanenko and Vukobratovic were replaced by one.

1.4 Recursive Lagrangian Dynamics

Hollerbach [3] realized that the recursive nature of the Newton-Euler formulation that made it so efficient could be achieved with the Lagrangian formulation as well. Starting with the original results of Uicker and Kahn, he developed forward and backward recurrence relations for the terms in (1.2) that allow the generalized forces to be computed in linear time. The result for a 6-link manipulator still required about 5 times the number of adds and multiplies as the Luh formulation, but this is about 15 times better than direct evaluation of (1.2).

Hollerbach also realized that the use of 4×4 rotation/translation matricies to represent the position and motion of the chain led to inefficiencies in the calculation. He reformulated the Lagrangian dynamics in terms of pure rotation matricies to specify the orientation of the links, and displacement vectors to specify their position. This reformulation resulted in an additional factor of 2 savings in adds and multiplies, bringing the Lagrangian formulation to within roughly a factor of $2\frac{1}{2}$ of Luh's Newton-Euler formulation.

1.5 The Importance of the Representation of Angular Velocity

Hollerbach used a rotation matrix W_i to specify the orientation of link i of the kinematic chain. W_i transforms the components of a vector with respect to a coordinate system fixed in link i to its components with respect to a fixed (inertial) coordinate system. The angular motion of link i is represented by the time derivatives of W_i: $\dot{W}_i$ and $\ddot{W}_i$.

Although W_i, $\dot{W}_i$, and $\ddot{W}_i$ each have nine components, orientation has only three degrees of freedom and thus only three of the components are independent. An equivalent representation for the angular motion of a link is the angular velocity vector ω_i and the angular acceleration vector $\dot{\omega}_i$, which have been used in all of the Newton-Euler formulations reported here, and which contain no redundant information. Unfortunately, there in no "orientation vector" corresponding to W_i; the Euler angles or equivalent may be used instead, although we will not need to do so here.

In the next section we explore in detail the relationship between ω and $\dot{W}$, and the resulting descriptions of rotational dynamics. In the following section we show that a Lagrangian formulation based on ω instead of $\dot{W}$ leads to exactly the same computation as the Newton-Euler formulation. This result is hardly surprising, since both methods must give the same numerical solution, and we are now starting with exactly the same quantities. The significance of this result is not just that it doesn't matter which formulation one uses. Rather, it shows what the real issues are if one is interested in efficiency: structure of

the computation and choice of representation.

2. Comparison of Rotational Dynamics with ω and $\dot{W}$

It was pointed out above that the angular motion of a rigid body could be described equally well by either the angular velocity vector ω or the derivative of a rotation matrix, $\dot{W}$. Obviously, scalar quantities such as kinetic energy must be independent of the representation chosen, but higher rank tensor quantities need not be, and expressions for any quantity will be different in form. Furthermore, neither representation is clearly better in all cases. Use of ω may yield a more efficient computation, but it has the disadvantage that there is no "angular position vector" that it is the derivative of. Therefore, it is interesting and useful to compare the description of rotational dynamics that results from different choices of representation, and to develop formulas that allow one to switch between representations. That is the main purpose of this section, although in making this comparison we will also get expressions for rotational inertia and kinetic energy which are needed in the next section.

2.1 $\dot{W}$

We start by defining $\dot{W}$ more formally and introducing some conventions that are needed below. We assume that the reader is familiar with the properties of the angular velocity vector ω and rotating reference frames, which have been discussed in many texts (see, for example, [5]).

Let $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$ be any fixed (inertial) orthonormal basis, and let $\{\hat{e}'_1, \hat{e}'_2, \hat{e}'_3\}$ be any orthonormal basis attached to a rotating rigid body. We will always use primes to indicate rotating basis vectors or the components of a vector with respect to such a basis. Thus, if v is any vector we have:

$$v = \sum_i v_i \hat{e}_i = \sum_i v'_i \hat{e}'_i.$$

From now on we will drop the summation sign and use the Einstein summation convention, that is, indices that appear twice in any term imply a summation of that term over all values of the indices (i.e. $\{1, 2, 3\}$).

Using the above conventions we can write:

$$\begin{aligned} v &= v'_j \hat{e}'_j \\ v \cdot \hat{e}_i &= \hat{e}_i \cdot \hat{e}'_j v'_j \\ v_i &= W_{ij} v'_j, \qquad W_{ij} = \hat{e}_i \cdot \hat{e}'_j \\ \dot{W}_{ij} &= \frac{dW_{ij}}{dt}. \end{aligned} \tag{2.1}$$

Note also that since the inverse of W is its transpose, we have:

$$W_{ik} W^T_{kj} = W_{ik} W_{jk} = \delta_{ij}, \tag{2.2}$$

where δ_{ij} are the components of the identity tensor δ.

2.2 The Cross Product Operation

We must now briefly discuss the cross-product operation, which is indispensible when dealing with rotation in three dimensions, and which must be used unambiguously with second rank tensors as well as vectors. We assume that the standard geometric definition of the cross product of two vectors is known to the reader.

The cross product may be viewed as a function of two vectors that produces a third vector:

$$\underline{a} \times \underline{b} = \underline{c} = P(\underline{a}, \underline{b}).$$

Alternatively, we may view the cross product as a function of three vectors that produces a scalar, the so-called scalar triple product:

$$\underline{a} \times \underline{b} \cdot \underline{c} = s = P(\underline{a}, \underline{b}, \underline{c}).$$

The function P is:

- Linear in $\underline{a}$, $\underline{b}$, and $\underline{c}$,
- Independent of the choice of basis vectors (since it is defined geometrically).

Therefore, P is a rank three tensor[1].

To find the components of P according to some right-handed basis $\{\hat{e}_1, \hat{e}_2, \hat{e}_3\}$, we simply apply the function to the basis vectors:

$$\begin{aligned} P_{ijk} &= P(\hat{e}_i, \hat{e}_j, \hat{e}_k) \\ &= \hat{e}_i \times \hat{e}_j \cdot \hat{e}_k \\ &= \begin{cases} 1, & \text{if } ijk \text{ is an even permutation of 123,} \\ -1, & \text{if } ijk \text{ is an odd permutation of 123,} \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

From this it can be seen that P is totally anti-symmetric—swapping any two indices changes the sign (but rotating the indices has no effect). To illustrate how P is used, here are the formulas for the cross-product of two vectors, and the scalar triple product, in coordinates:

$$[\underline{a} \times \underline{b}]_i = P_{ijk} a_j b_k \qquad \underline{a} \cdot \underline{b} \times \underline{c} = P_{ijk} a_i b_j c_k.$$

There are two very useful identities associated with the components of P:

$$P_{ijk} P_{ijl} = 2\delta_{kl}, \tag{2.3}$$

and

$$P_{ijk} P_{ilm} = \delta_{jl}\delta_{km} - \delta_{jm}\delta_{kl}. \tag{2.4}$$

[1] Technically, a *pseudotensor*, since it requires consistent use of right- or left-handed coordinate systems, but not both.

2.3 The Relationship Between ω and $\dot{W}$

We can now find the explicit relationship between ω and $\dot{W}$. That relationship will be seen to be position (i.e. orientation) dependent, and the position will always be represented by the rotation matrix W.

Let v be any vector rotating with angular velocity ω. We can derive the components of $\dot{v}$ in the following two ways:

1) $\dot{v} = \omega \times v$ 2) $v_i = W_{ij} v'_j$

$\dot{v}_i = P_{ikl}\omega_k v_l$ $\dot{v}_i = \dot{W}_{ij} v'_j$

Equating the two gives:

$$\dot{W}_{ij} v'_j = P_{ikl}\omega_k v_l = P_{ikl}\omega_k W_{lj} v'_j.$$

Since this must hold for any v, it is clear that

$$\dot{W}_{ij} = P_{ikl}\omega_k W_{lj} \tag{2.5}$$

(or in vector notation, $\dot{W} = \omega \times W$). The inverse relation may be found from this and equations (2.2) and (2.3) as follows:

$$\dot{W}_{ij} = P_{ikl}\omega_k W_{lj}$$

$$W_{mj}\dot{W}_{ij} = P_{ikl}\omega_k \delta_{lm} = P_{ikm}\omega_k$$

$$P_{inm}W_{mj}\dot{W}_{ij} = P_{ikm}P_{inm}\omega_k = 2\delta_{ki}\omega_k = 2\omega_i.$$

Making an appropriate change of dummy indices, we get

$$\omega_i = \tfrac{1}{2} P_{ijk} W_{jl} \dot{W}_{kl}, \tag{2.6}$$

to which there is no corresponding vector notation known to the author.

2.4 Rotational Inertia and Kinetic Energy

This section has three purposes: it gives one example of how a dynamic quantity (rotational kinetic energy) can be expressed in terms of either ω or $\dot{W}$, it provides one way of defining the inertia tensor (the

definition will depend on whether ω or $\dot{W}$ is used), and it provides the expression for rotational kinetic energy that is needed for the Lagrangian formulation of the next section.

Let r be a vector from the center of mass of a rigid body to a small volume element of mass dm. The velocity and kinetic energy of the volume element (due to rotation of the body) can be written as:

$$v = \dot{r}$$
$$dK = \tfrac{1}{2} v \cdot v \, dm.$$

We will express dK in terms of both $\dot{W}$ and ω and integrate over the body to get the total rotational kinetic energy. The definitions of the inertia tensor will fall out of the derivation. Components of vectors are with respect to any right-handed orthonormal basis, primed for rotating and unprimed for fixed, as above. First, using $\dot{W}$:

$$\begin{aligned} r_i &= W_{ij} r'_j \\ v_i &= \dot{W}_{ij} r'_j \\ dK &= \tfrac{1}{2}(\dot{W}_{ij} r'_j)(\dot{W}_{ik} r'_k) dm \\ K &= \tfrac{1}{2} \dot{W}_{ij} \dot{W}_{ik} \int_V r'_j r'_k dm \\ \text{let} \quad J'_{jk} &= \int_V r'_j r'_k dm \qquad (2.7) \\ K &= \tfrac{1}{2} \dot{W}_{ij} J'_{jk} \dot{W}_{ik} \\ &= \tfrac{1}{2} tr(\dot{W} J' \dot{W}^T). \qquad (2.8) \end{aligned}$$

The last expression is in matrix form for the benifit of readers who are more familiar with that notation.

Now we repeat the derivation, using ω instead:

$$
\begin{aligned}
v &= \omega \times r \\
dK &= \tfrac{1}{2}(\omega \times r)\cdot(\omega \times r)dm \\
&= \tfrac{1}{2}[(\omega\cdot\omega)(r\cdot r) - (\omega\cdot r)(\omega\cdot r)]dm \\
&= \tfrac{1}{2}[\omega_i\omega_i r^2 - \omega_i r_i \omega_j r_j]dm \\
&= \tfrac{1}{2}[\omega_i\omega_j\delta_{ij}r^2 - \omega_i\omega_j r_i r_j]dm \\
&= \tfrac{1}{2}\omega_i\omega_j(r^2\delta_{ij} - r_i r_j)dm \\
K &= \tfrac{1}{2}\omega_i\omega_j\int_V (r^2\delta_{ij} - r_i r_j)dm \\
\text{let}\quad I_{ij} &= \int_V (r^2\delta_{ij} - r_i r_j)dm \qquad (2.9)\\
K &= \tfrac{1}{2}\omega_i\omega_j I_{ij} = \tfrac{1}{2}\omega\cdot I\cdot\omega \qquad (2.10)
\end{aligned}
$$

From equations (2.7) and (2.9) the relationship between the two inertia tensors can be seen:

$$I = tr(J)\delta - J, \qquad J = \tfrac{1}{2}tr(I)\delta - I. \qquad (2.11)$$

Note that since I and J have been defined above by their components, they have not actually been shown to be tensors. The proof is simple and can be supplied by the reader.

From equations (2.8) and (2.10), and a definition of a generalized coordinate q, the rotational contribution to the corresponding generalized force can be found from the Lagrange equation. This was done by Hollerbach [3] for $\dot{W}$, and is done in the next section for ω. We summarize the results here for comparison:

angular velocity	ω	$\dot{W}$
kinetic energy	$\tfrac{1}{2}\omega\cdot I\cdot\omega$	$\tfrac{1}{2}tr(\dot{W}J\dot{W}^T)$
generalized force	$[\dot{\omega}\cdot I + \omega\times(I\cdot\omega)]\cdot\dfrac{\partial\omega}{\partial\dot{q}}$	$tr\left(\dfrac{\partial W}{\partial q}J\ddot{W}^T\right)$

While these expressions have been derived independently, their equality can be verified by direct substitution using equations (2.4), (2.5), (2.6), and (2.11).

3. Lagrangian Dynamics Using ω Instead of $\dot{W}$

In this section we give the details of the Lagrangian formulation based on the angular velocity vector $\underset{\sim}{\omega}$. The generalized forces are derived for any open-loop kinematic chain, and the results are interpreted and compared to the Newton-Euler formulation of Luh [4].

One feature that distinguishes the present formulation from previous ones is that expressions for the generalized forces are derived without defining a single coordinate system. All quantities are expressed in terms of geometric objects (tensors) and geometric operations (vector addition, dot and cross product). It is only at the end, when a computation must be derived from these expressions, that coordinate systems must be defined, so that the various quantities can be measured and represented in a computer program. At this point, expressions may be evaluated in any convenient right-handed orthonormal coordinate system, provided *that this is done so in a consistent manner*. In practice, the method presented by Luh [4] is probably most efficient.

3.1 The Derivation of the Generalized Forces

In this section we write down the total kinetic energy of a kinematic chain and apply the Lagrange equation to derive the generalized forces. The potential energy term due to gravity is omitted here as mentioned in section 1.2, although this and other minor details are taken care of below.

The total kinetic energy of a rigid body is the sum of the energy due to the motion of the center of mass and the energy due to rotation about the center of mass [5]:

$$K = \tfrac{1}{2}m(v)^2 + \tfrac{1}{2}\underset{\sim}{\omega}\cdot\underset{\sim}{I}\cdot\underset{\sim}{\omega}.$$

To get the total energy of a chain, sum over all of the links:

$$K = \sum_{i=1}^{n} \left[\tfrac{1}{2} m_i (v_i)^2 + \tfrac{1}{2} \omega_i \cdot I_i \cdot \omega_i \right]. \tag{3.1}$$

We may now compute the derivatives required by the Lagrange equation (1.1).

$$\frac{\partial K}{\partial \dot{q}_j} = \sum_i \left[m_i v_i \cdot \frac{\partial v_i}{\partial \dot{q}_j} + \omega_i \cdot I_i \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} \right] \tag{3.2}$$

The first term of (3.2) follows directly from the chain rule. For the second term, note that while the inertia tensor I_i is a function of position, it is independent of any joint velocity $\dot{q}_j$. Furthermore, since I is symmetric,

$$\tfrac{1}{2} \frac{\partial \omega_i}{\partial \dot{q}_j} \cdot I_i \cdot \omega_i = \tfrac{1}{2} \omega_i \cdot I_i \cdot \frac{\partial \omega_i}{\partial \dot{q}_j}.$$

Straightforward application of the rules for differentiating products gives:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_j}\right) = \sum_i \Bigg[& m_i \dot{v}_i \cdot \frac{\partial v_i}{\partial \dot{q}_j} + m_i v_i \cdot \frac{d}{dt}\left(\frac{\partial v_i}{\partial \dot{q}_j}\right) \\ & + \dot{\omega}_i \cdot I_i \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} + \omega_i \cdot \dot{I}_i \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} + \omega_i \cdot I_i \cdot \frac{d}{dt}\left(\frac{\partial \omega_i}{\partial \dot{q}_j}\right) \Bigg]. \end{aligned} \tag{3.3}$$

It is shown in Appendix A that:

$$\frac{d}{dt}\left(\frac{\partial v_i}{\partial \dot{q}_j}\right) = \frac{\partial v_i}{\partial q_j} \tag{3.4}$$

$$\omega_i \cdot \dot{I}_i \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} = \omega_i \times (I_i \cdot \omega_i) \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} \tag{3.5}$$

$$\frac{d}{dt}\left(\frac{\partial \omega_i}{\partial \dot{q}_j}\right) = \frac{\partial \omega_i}{\partial q_j} + \omega_i \times \frac{\partial \omega_i}{\partial \dot{q}_j} \tag{3.6}$$

$$\omega_i \cdot I_i \cdot \left(\omega_i \times \frac{\partial \omega_i}{\partial \dot{q}_j}\right) = -\omega_i \times (I_i \cdot \omega_i) \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} \tag{3.7}$$

Substituting (3.4)-(3.7) into (3.3) gives:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_j}\right) = \sum_i \left[m_i \dot{v}_i \cdot \frac{\partial v_i}{\partial \dot{q}_j} + m_i v_i \cdot \frac{\partial v_i}{\partial q_j} + \dot{\omega}_i \cdot \underline{I}_i \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} + \omega_i \cdot \underline{I}_i \cdot \frac{\partial \omega_i}{\partial q_j} \right]. \tag{3.8}$$

Now for the final term in the Lagrange equation:

$$\frac{\partial K}{\partial q_j} = \sum_i \left[m_i v_i \cdot \frac{\partial v_i}{\partial q_j} + \omega_i \cdot I_i \cdot \frac{\partial \omega_i}{\partial q_j} + \tfrac{1}{2}\omega_i \cdot \frac{\partial I_i}{\partial q_j} \cdot \omega_i \right] \tag{3.9}$$

This is very similar to (3.2), except for the appearence of a term due to the position dependence of the inertia tensor. It is shown in Appendix A that:

$$\tfrac{1}{2}\omega_i \cdot \frac{\partial \underline{I}_i}{\partial q_j} \cdot \omega_i = -\omega_i \times (\underline{I}_i \cdot \omega_i) \cdot \frac{\partial \omega_i}{\partial \dot{q}_j}. \tag{3.10}$$

Putting it all together:

$$\tau_j = \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_j}\right) - \frac{\partial K}{\partial q_j}$$
$$\sum_i \left(m_i \dot{v}_i \cdot \frac{\partial v_i}{\partial \dot{q}_j} + [\dot{\omega}_i \cdot I_i + \omega_i \times (I_i \cdot \omega_i)] \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} \right) \tag{3.11}$$

3.2 Comparison with the Newton-Euler Formulation

We have derived the generalized forces in a very general way, without unnecessary details like coordinate systems, link and joint numbering conventions, and other conventions needed to specify the geometry of the manipulator. In order to interpret the result and compare it to the Newton-Euler formulation, however, it is finally necessary to make some of these definitions. Such details are not the point of this paper, and will be kept to a minimum.

The links of the manipulator are numbered consecutively from the base to the tip, as are the joints that connect them. The base is considered to be link 0, while the terminal link is numbered link n. The

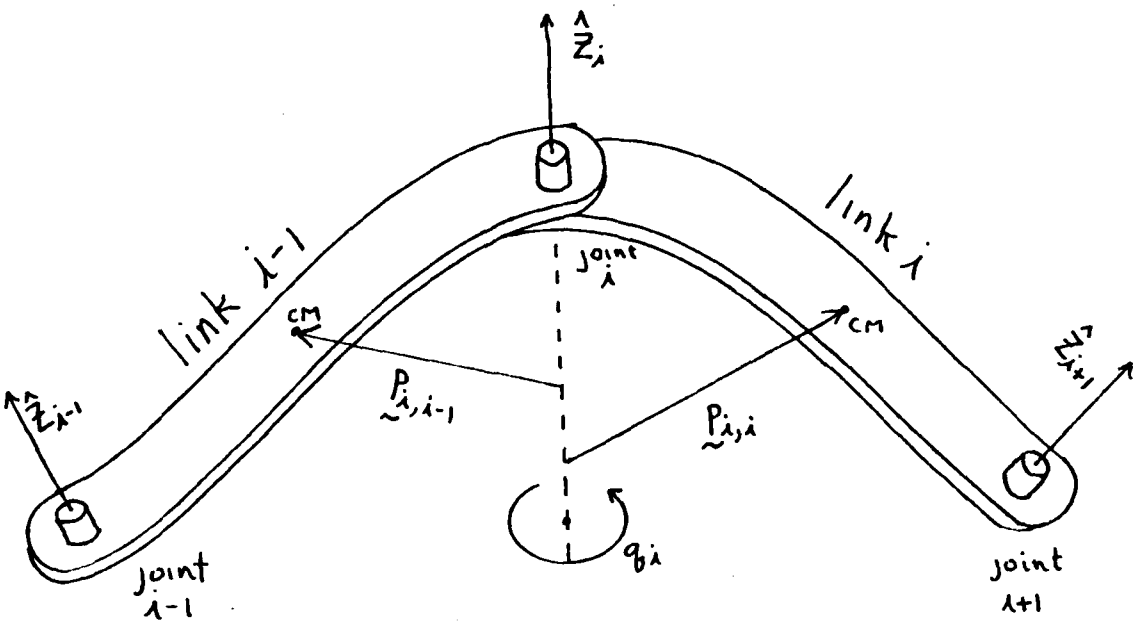


Figure 3.1. Link and joint numbering, and other conventions for open-loop kinematic chains.

joints are numbered 1 thru n, joint 1 connecting link 1 to the base. Thus joint i connects links $i-1$ and i; link i is bounded by joints i and $i+1$, as shown in figure 3.1.

If joint i is rotational, the joint variable q_i measures the angle of rotation from some (arbitrary for the present discussion) reference point; if it is translational, q_i measures the sliding distance. The unit vector $\hat{z}_i$ is attached to joint i and points along the axis of rotation for rotary joints or along the sliding axis for sliding joints. Note that for rotary joints, q_i must be measured in a right-hand sense about $\hat{z}_i$. Finally, let $p_{j,i}$ be a position vector that points from anywhere along the axis of joint j to the center of mass of link i. (Note: These definitions of $\hat{z}_i$ and $p_{j,i}$ are non-standard, and are clearly too ambiguous to be used in practice. They are, however, all that is needed to understand equation (3.11)).

One of the first things one notices when looking at equation (3.11) is that it contains the Newton-

Euler expressions for the total vector force and moment acting on a rigid body. In fact we can rewrite (3.11) as:

$$\tau_j = \sum_i \left(\underset{\sim}{F}_i \cdot \frac{\partial \underset{\sim}{v}_i}{\partial \dot{q}_j} + \underset{\sim}{N}_i \cdot \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} \right). \tag{3.12}$$

Thus we are very interested in the vectors $\partial \underset{\sim}{v}_i / \partial \dot{q}_j$ and $\partial \underset{\sim}{\omega}_i / \partial \dot{q}_j$ that the force $\underset{\sim}{F}_i$ and moment $\underset{\sim}{N}_i$ are projected onto. These vectors specify the dependence of the linear and angular velocity of link i on the joint velocity of joint j. For $j > i$ there is no such dependence, and so these vectors are 0. This means that the summation in equation (3.12) can be taken from $i = j$ to $i = n$, instead of over all i.

For $j \leq i$, we note that the linear and angular motion of link i may be written as the vector sum of contributions due to the relative motion of the previous links at the joints:

$$\underset{\sim}{v}_i = \sum_{j=1}^{i} \begin{cases} (\dot{q}_j \hat{z}_j) \times \underset{\sim}{p}_{j,i}, & \text{if joint } j \text{ is rotational;} \\ \dot{q}_j \hat{z}_j, & \text{if joint } j \text{ is translational.} \end{cases} \tag{3.13}$$

$$\omega_i = \sum_{j=1}^{i} \begin{cases} \dot{q}_j \hat{z}_j, & \text{if joint } j \text{ is rotational;} \\ \underset{\sim}{0}, & \text{if joint } j \text{ is translational.} \end{cases} \tag{3.14}$$

Differentiating gives:

$$\frac{\partial \underset{\sim}{v}_i}{\partial \dot{q}_j} = \begin{cases} \hat{z}_j \times \underset{\sim}{p}_{j,i}, & \text{if joint } j \text{ is rotational;} \\ \hat{z}_j, & \text{if joint } j \text{ is translational.} \end{cases} \tag{3.15}$$

$$\frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} = \begin{cases} \hat{z}_j, & \text{if joint } j \text{ is rotational;} \\ \underset{\sim}{0}, & \text{if joint } j \text{ is translational.} \end{cases} \tag{3.16}$$

If joint j is translational, the joint force can now be written:

$$\tau_j = \hat{z}_j \cdot \sum_{i=j}^{n} F_i.$$

If joint j is rotary, the torque is:

$$\tau_j = \hat{z}_j \cdot \sum_{i=j}^{n} (p_{j,i} \times F_i + N_i).$$

Thus, to compute the generalized forces we first iterate from the base to the tip to compute $\dot{v}_i$, ω_i, and $\dot{\omega}_i$, and then iterate from the tip to the base using the above relations to compute the forces. Referring to [4], it can be seen that this is exactly the computation specified by Luh that was derived from the Newton-Euler approach.

A few minor points still need to be cleared up, however. First, we are still free to choose a coordinate system or systems in which to evaluate these expressions. The method presented in [4] is probably best, where quantities associated with a given link (such as the inertia tensor) are expressed in a coordinate system attached to that link, and then are transformed to the coordinates of the previous link as the iteration proceeds.

Second, we must say something about the gravity term which we have thus far ignored. We could include it in the Lagrange equation in the standard way, as a position dependent potential energy term. This is equivalent to its inclusion in the Newton-Euler formulation as described in step 3 of section 1.3. Perhaps a better way was also discussed by Luh -- instead of considering the base as fixed, give it a vector acceleration equal to that due to gravity. Both methods will give the same numerical result, but Luh's is probably more efficient since the effect of gravity is computed only once.

Third is the sliding friction forces produced in the joints that Luh includes in his equations. These, however, are simply computed based on the joint velocity and are added directly to the joint generalized force. This clearly can be done no matter how the dynamics are formulated.

Finally, the Newton-Euler dynamics includes the solution to the problem of the statics of the manipulator, that is the effect of the external force and moment acting on the terminal link. This can be included in the Lagrangian formulation by supposing an additional link attached to the terminal link,

whose motion is defined so as to produce the required external forces. This is very much like including gravity by specifying the acceleration of the base rather than the equivalent forces.

A. Details for the Lagrangian Formulation

In this Appendix we supply the details that were omitted from the derivation of the generalized forces presented in section 2.1.

For equation (3.4), let r_i be a position vector from any fixed origin to the center of mass of link i, so that $v_i = \dot{r}_i$. Then we have:

$$\frac{\partial v_i}{\partial \dot{q}_j} = \frac{\partial \dot{r}_i}{\partial \dot{q}_j} = \frac{\partial}{\partial \dot{q}_j} \sum_k \frac{\partial r_i}{\partial q_k} \dot{q}_k = \frac{\partial r_i}{\partial q_j} \tag{A.1}$$

$$\frac{d}{dt}\left(\frac{\partial v_i}{\partial \dot{q}_j}\right) = \frac{d}{dt}\left(\frac{\partial r_i}{\partial q_j}\right) = \sum_k \frac{\partial^2 r_i}{\partial q_j \partial q_k} \dot{q}_k = \frac{\partial}{\partial q_j} \sum_k \frac{\partial r_i}{\partial q_k} \dot{q}_k$$

$$= \frac{\partial}{\partial q_j} \dot{r}_i = \frac{\partial v_i}{\partial q_j}$$

Equation (3.5):

$$\begin{aligned}
\omega_i \cdot \underline{\underline{I}}_i \cdot \frac{\partial \underline{\omega}_i}{\partial \dot{q}_j} &= \underline{\omega}_i \cdot (\underline{\omega}_i \times \underline{\underline{I}}_i - \underline{\underline{I}}_i \times \underline{\omega}_i) \cdot \frac{\partial \underline{\omega}_i}{\partial \dot{q}_j} \\
&= \underline{\omega}_i \cdot (\underline{\omega}_i \times \underline{\underline{I}}_i) \cdot \frac{\partial \underline{\omega}_i}{\partial \dot{q}_j} - \underline{\omega}_i \cdot (\underline{\underline{I}}_i \times \underline{\omega}_i) \cdot \frac{\partial \underline{\omega}_i}{\partial \dot{q}_j} \\
&= \underline{\omega}_i \times \underline{\omega}_i \cdot \underline{\underline{I}}_i \cdot \frac{\partial \underline{\omega}_i}{\partial \dot{q}_j} - (\underline{\omega}_i \cdot \underline{\underline{I}}_i) \times \underline{\omega}_i \cdot \frac{\partial \underline{\omega}_i}{\partial \dot{q}_j} \\
&= \underline{\omega}_i \times (\underline{\underline{I}}_i \cdot \underline{\omega}_i) \cdot \frac{\partial \underline{\omega}_i}{\partial \dot{q}_j}.
\end{aligned}$$

Equation (3.7):

$$\begin{aligned}
\omega_i \cdot I_i \cdot \left(\omega_i \times \frac{\partial \omega_i}{\partial \dot{q}_j}\right) &= (\omega_i \cdot I_i) \times \omega_i \cdot \frac{\partial \omega_i}{\partial \dot{q}_j} \\
&= -\omega_i \times (I_i \cdot \omega_i) \cdot \frac{\partial \omega_i}{\partial \dot{q}_j}.
\end{aligned}$$

For equation (3.10): Following a procedure similar to that used in (A.1) above, it can be shown that:

$$\frac{\partial \underset{\sim}{I}_i}{\partial q_j} = \frac{\partial \dot{\underset{\sim}{I}}_i}{\partial \dot{q}_j}.$$

Thus we have:

$$\begin{aligned}\frac{\partial \underset{\sim}{I}_i}{\partial q_j} &= \frac{\partial}{\partial \dot{q}_j}(\underset{\sim}{\omega}_i \times \underset{\approx}{I}_i - \underset{\approx}{I}_i \times \underset{\sim}{\omega}_i) \\ &= \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} \times \underset{\approx}{I}_i - \underset{\approx}{I}_i \times \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j}.\end{aligned}$$

Therefore,

$$\begin{aligned}\tfrac{1}{2}\underset{\sim}{\omega}_i \cdot \frac{\partial \underset{\sim}{I}_i}{\partial q_j} \cdot \underset{\sim}{\omega}_i &= \tfrac{1}{2}\underset{\sim}{\omega}_i \cdot \left(\frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} \times \underset{\approx}{I}_i\right) \cdot \underset{\sim}{\omega}_i - \tfrac{1}{2}\underset{\sim}{\omega}_i \cdot \left(\underset{\approx}{I}_i \times \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j}\right) \cdot \underset{\sim}{\omega}_i \\ &= \tfrac{1}{2}\underset{\sim}{\omega}_i \cdot \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} \times (\underset{\approx}{I}_i \cdot \underset{\sim}{\omega}_i) - \tfrac{1}{2}(\underset{\sim}{\omega}_i \cdot \underset{\approx}{I}_i) \times \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} \cdot \underset{\sim}{\omega}_i \\ &= -\tfrac{1}{2}\underset{\sim}{\omega}_i \times (\underset{\approx}{I}_i \cdot \underset{\sim}{\omega}_i) \cdot \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} - \tfrac{1}{2}\underset{\sim}{\omega}_i \times (\underset{\approx}{I}_i \cdot \underset{\sim}{\omega}_i) \cdot \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j} \\ &= -\underset{\sim}{\omega}_i \times (\underset{\approx}{I}_i \cdot \underset{\sim}{\omega}_i) \cdot \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j}.\end{aligned}$$

The final (and trickiest) proof is equation (3.6):

$$\frac{d}{dt}\left(\frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j}\right) = \frac{\partial \underset{\sim}{\omega}_i}{\partial q_j} + \underset{\sim}{\omega}_i \times \frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j}.$$

We will need equations (3.14) and (3.16), and the conventions of section 3.2 (see figure 3.1), except that for convenience we will take $\hat{z}_k$ to be $\underset{\sim}{0}$ if joint k is translational.

First note that if joint j is translational, or if $j > i$, $\underset{\sim}{\omega}_i$ is independent of both q_j and $\dot{q}_j$, so that both sides of the equation are identically zero. Now for $j \leq i$, since $\hat{z}_j$ is attached to joint j and therefore link j, we may write:

$$\frac{d}{dt}\left(\frac{\partial \underset{\sim}{\omega}_i}{\partial \dot{q}_j}\right) = \frac{d\hat{z}_j}{dt} = \underset{\sim}{\omega}_j \times \hat{z}_j. \qquad (A.2)$$

By considering the rotation of a vector by some angle about a given axis, it can be seen that:

$$\frac{\partial \hat{z}_k}{\partial q_j} = \hat{z}_j \times \hat{z}_k \quad (j \leq k). \qquad (A.3)$$

Now we have everything we need for the proof. Starting with (A.2),

$$\begin{aligned}
\frac{d}{dt}\left(\frac{\partial \omega_i}{\partial \dot{q}_j}\right) &= \omega_j \times \hat{z}_j \\
&= (\omega_i - \sum_{k=j+1}^{i} \dot{q}_k \hat{z}_k) \times \hat{z}_j \\
&= \omega_i \times \hat{z}_j + \sum_{k=j+1}^{i} \dot{q}_k \hat{z}_j \times \hat{z}_k \\
&= \omega_i \times \hat{z}_j + \sum_{k=1}^{i} \dot{q}_k \frac{\partial \hat{z}_k}{\partial q_j} \\
&= \omega_i \times \hat{z}_j + \frac{\partial}{\partial q_j} \sum_{k=1}^{i} \dot{q}_k \hat{z}_k \\
&= \omega_i \times \frac{\partial \omega_i}{\partial \dot{q}_j} + \frac{\partial \omega_i}{\partial q_j}.
\end{aligned}$$

Acknowledgements

I would like to thank John Hollerbach, Mike Brady, and Berthold Horn for many useful discussions and comments, and Frank Morgan for teaching me tensor algebra.

B-104

References

[1] J.J. Uicker, "On the Dynamic Analysis of Spatial Linkages Using 4x4 Matricies," PhD Thesis, Northwestern University, August 1965.

[2] M.E. Kahn, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," Stanford Artificial Intelligence Project Memo AIM-106, December 1969.

[3] J.M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparitive Study of Dynamics Formulation Complexity," *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 730-736, Nov. 1980.

[4] J. Luh, M. Walker and R. Paul, "On-line Computational Scheme for Mechanical Manipulators," 2nd IFAC/IFIP Symposium on Information Control Problems in Manufacturing Technology, Stuttgart, Germany, October 22-24, 1979.

[5] K.R. Symon, *Mechanics*, 3rd edition, Addison-Wesley, 1971.

[6] Y. Stepanenko and M. Vukobratovic, "Dynamics of Articulated Open-Chain Active Mechanisms," *Math. Biosc.*, vol. 28, pp. 137-170, 1976.

[7] D.E. Orin, R.B. McGhee, M. Vukobratovic and G. Hartoch, "Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods," *Math. Biosc.*, vol. 43, pp. 107-130, 1979.

[8] W.W. Armstrong, "Recursive Solution to the Equations of Motion of an N-Link Manipulator," *Proc. 5th World Congress on Theory of Machines and Mechanisms*, vol. 2, pp. 1343-1346, July 1979.


DAT
FILM